الفصل الخامس

حلول المعادلات التفاضلية على MATLAB

المعادلات التفاضلية هي معادلات ذات أهمية خاصة لدى المهندسين، والفيزيائيين والرياضيين، لأن ظواهر عديدة فيزيائية، بيولوجية، وكيميائية، واقتصادية تُمثل رياضياً بهذه المعادلات. في هذا الفصل نوضح كيفية استخدام MATLAB لإيجاد حلول عددية لبعض أنواع المعادلات التفاضلية.

(٥,١) مقدمة في المعادلات التفاضلية

المعادلة التفاضلية العادية Ordinary differential equation هي معادلة تتضمن دالة $y = f(t)$ في متغير t ومشتقاتها $y', y'', \ldots, y^{(n)}$ على شكل:

$$F(t, y, y', y'', \ldots, y^{(n)}) = 0$$

وتعرّف رتبة المعادلة التفاضلية order بأنها رتبة المشتق ذي الرتبة العليا الذي يظهر فيها. تحسب درجة المعادلة degree بأخذ درجة الحد الأعلى رتبة. وتسمى المعادلة التفاضلية خطية linear إذا كانت y وجميع مشتقاتها من الدرجة الأولى ومعامل y

وجميع معاملات مشتقاتها بمتغير واحد t أو ثابت ، وإذا كانت المعادلات التفاضلية تحتوي على دالة في متغيرين أو أكثر وتفاضل جزئي ، فتسمى معادلات تفاضلية جزئية partial differential equation. نطلق على الدالة التي تحقق المعادلة التفاضلية حل المعادلة التفاضلية solution ، فمثلاً :

$$y'+y''=0$$

هي معادلة تفاضلية عادية خطية من الرتبة الثانية والدرجة الأولى ، أما الحل فهو دالة على شكل $y=e^{-t}$.

هناك معادلات تفاضلية بقيمة ابتدائية معطاة $y(t_0) = y_0$ وتسمى معادلات تفاضلية بقيم ابتدائية Initial value problem وأخرى بقيم عند حدود المجال ، وتسمى Boundary value problems. كما أن هناك طرقاً تحليلية للتوصل للحل الفعلي للمعادلات التفاضلية ، ولكن بعضاً من المعادلات لا يوجد لها حل فعلي ، أو أن الحل الفعلي يصعب التعامل معه ، لذلك نلجأ إلى طرق عددية للحصول على حل للمعادلة. وهذا يعني أننا نقدر الحل المتصل للمعادلة بقيم تقريبية متقطعة تعطي قيمة دالة الحل عند نقاط معينة في المجال تبدأ بنقطة ابتدائية وتنتهي عند نقطة نهائية معطاة. فنقرب $y(t_i)$ بالقيمة y_i للقيم t_i بحيث إن $t_i = t_0 + ih$ ، h طول الفترة و $i= 0,1,2,...n$.

في هذا الفصل سنوضح كيفية استخدام MATLAB في حل معادلات تفاضلية بطرق عددية ، كما سنعطي دوال جاهزة في MATLAB تحسب الحلول للمعادلات التفاضلية في البيئة المعتادة أو في البيئة الرمزية syms .

(٥,٢) طريقة أويلر Euler Method

لحل معادلة تفاضلية بقيمة ابتدائية توجد طرق ذات خطوة واحدة وأخرى بعدة

خطوات. تُعد طريقة أويلر *Euler* من طرق الخطوة الواحدة، ولا تحتاج لقيمة ابتدائية سوى المعطاة في المعادلة التفاضلية. و هي من أبسط الطرق العددية المباشرة لحل معادلة تفاضلية عادية من الرتبة الأولى، مثل:

$$y' = \frac{dy}{dt} = f(t,y) \qquad a \le t \le b, \qquad y(a) = y_0$$

لاستنتاج صيغة أويلر نستخدم سلسلة تايلور على دالة الحل $y(t)$:

$$y(t_i + h) = y(t_i) + y'(t_i)h + y''(\mu)h^2/2$$

لتأخذ طريقة أويلر الصيغة :

$$y(t_{i+1}) = y(t_i) + hy'(t_i)$$

حيث إن $h = t_{i+1} - t_i$ و العدد μ بين $(t_i,\ t_{i+1})$ والخطأ برتبة h^2 و يقدر بـ $(h^2/2)y''(\mu)$.

لذا يمكن كتابة صيغة أويلر النهائية:

$$y_0 = y(a)$$
$$y_{i+1} = y_i + hf(t_i, y_i)$$

لكل $i=0,1,2,3,\ldots n-1$ و $y_i = y(t_i)$. وتنفذ طريقة Euler بالخوارزمية (٥،١):

```
function [tvals, yvals]=feuler(f,start,finish,startval,h)
s=(finish-start)/h+1;
y=startval;t=start;
yvals=startval;tvals=start;
for i=2:s
  y1=y+h*feval(f,t,y);
 t1=t+h;
  tvals=[tvals, t1];
yvals=[yvals, y1];
  t=t1;
  y=y1;
end
```

خوارزمية (٥,١)

مثال رقم (٥,١)

استخدم الخوارزمية (٥,١) لحل المعادلة التفاضلية $y' = xy + x$ بقيمة ابتدائية $y(0) = 0$ عند $0 \leq x \leq 1$ و $h=0.1$.

الحل :

بعد تخزين الدالة في m-file يدعى *fun* ندخل :

```
[x,y]=feuler('fun',0,1,0, 0.1);
>> plot(y,'r');
>> hold;
>> plot(-1+exp((x.^2)/2));
```

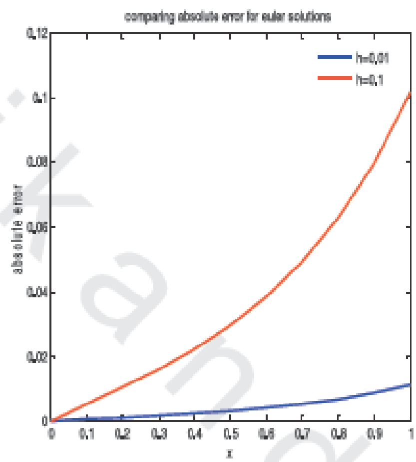
تظهر نتائج المتغيرات x , y ويمكن مقارنتها بالحل الفعلي $y(x) = e^{x^2/2} - 1$ عند نفس النقاط في الجدول رقم (٥,١).

الجدول رقم (١). نتائج مثال رقم (٥,١).

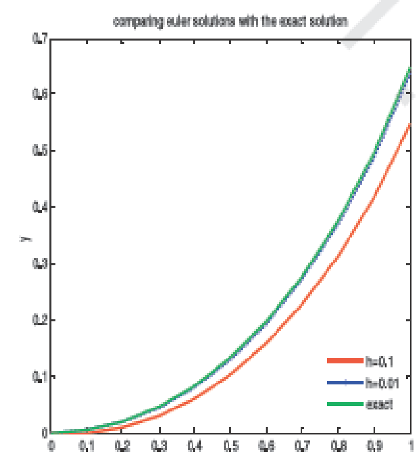
x	y	exact
0.1000	0	0.0050
0.2000	0.0100	0.0202
0.1000	0	0.0050
0.2000	0.0100	0.0202
0.3000	0.0302	0.0460
0.4000	0.0611	0.0833
0.5000	0.1036	0.1331
0.6000	0.1587	0.1972
0.7000	0.2283	0.2776
0.8000	0.3142	0.3771
1.0000	0.5471	0.6487

من الرسم نقارن الحل العددي الناتج من طريقة أويلر بالحل الفعلي ، ويمكن تحسين الحل بتصغير الخطوة h إلى $h= 0.01$ كما يظهر في الشكل رقم (٥,٢)، كما

يمكننا مقارنة الخطأ المطلق في الشكل رقم (٥.١) لنفس قيم *h*. ورغم أن طريقة أويلر يندر استخدامها عملياً(لكبر قيمة الخطأ) ، فإنه يمكن الاستفادة من بساطة استنتاجها لتوضيح الأساليب التي يتضمنها إنشاء بعض الطرائق الأكثر تقدماً.



الشكل رقم (٥,١). مقارنة الخطأ المطلق لحل أويلر التقريبي للمعادلة.



الشكل رقم (٥.٢). مقارنة الحل الفعلي وحل أويلر التقريبي للمعادلة.

(٥,٣) طريقة رونج كوتا Runge Kutta Method

تتكون طريقة رونج كوتا Runge-Kutta من مجموعة من الطرق العددية لحل المعادلات التفاضلية وهي عملية جداً وتعطي حلولاً بدقة عالية. ولطرائق رونج كوتا خطأ قطع محلي local truncation error ذو رتبة عالية، بينما لا تحتاج إلى حساب وتقدير مشتقات $f(x,y)$ مما يوجد في الطرق الأخرى.

هناك طرق لرونج كوتا من الدرجة الثانية second order Runge-Kutta methods مثل طريقة هيون Heun method وطريقة النقطة الوسيطة Midpoint method وطريقة أويلر المعدلة Modified Euler method . وتُعد هذه العائلة من طرق رونج كوتا سهلة البرمجة، وتستخدم أحياناً لإيجاد نقاط ابتدائية لبرامج أخرى. فمثلاً بفرض $y_0 = y(a)$ ولكل $i=0,1,2,3,\ldots n-1$

طريقة هيون تأخذ الصيغة العامة :

$$y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2)$$
$$k_1 = f(t_i, y_i) \quad , \mathrm{k}_2 = f(t_{i+1}, y_i + hk_1)$$

و صيغة طريقة النقطة الوسيطة :

$$y_{i+1} = y_i + hf(t_i + \frac{h}{2}, y_i + \frac{h}{2} f(t_i, y_i))$$

وطريقة أويلر المعدلة تأخذ الصيغة :

$$y_{i+1} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_i + hf(t_i, y_i))]$$

كما توجد طرق رونج كوتا من درجات أعلى مثل رونج كوتا الرابعة

الكلاسكية Classical RK4 ذات الخطأ برتبة $O(h^4)$:

$$
\begin{aligned}
&\textit{Runge Kutta 4}\\
&k_1 = hf(t_i, y_i)\\
&k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)\\
&k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right)\\
&k_4 = hf(t_i + h, y_i + k_3)\\
&y_{i+1} = y_i + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4]
\end{aligned}
$$

أو رونـــج كوتا ميرســـون RK Merson ذات الخطأ برتبة $O(h^5)$ [28] و تأخذ الصيغ التالية بفرض $y_0 = y(a)$ ولكل $i=0,1,2,3,\ldots n-1$:

$$
\begin{aligned}
&\textit{Runge Kutta Merson}\\
&k_1 = hf(t_i, y_i)\\
&k_2 = hf\left(t_i + \frac{h}{3}, y_i + \frac{k_1}{3}\right)\\
&k_3 = hf\left(t_i + \frac{h}{3}, y_i + \frac{k_2}{6} + \frac{k_1}{6}\right)\\
&k_4 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{8} + \frac{3k_3}{8}\right)\\
&k_5 = hf\left(t_i + h, y_i + \frac{k_1}{2} - \frac{3k_3}{2} + 2k_4\right)\\
&y_{i+1} = y_i + \frac{1}{6}[k_1 + 4k_4 + k_5]
\end{aligned}
$$

يمكن تطبيق الخوارزمية rgen (الملحق) [15] التي تسمح للمستخدم الاختيار ما بين الطريقتين المختلفتين من رونج كوتا عند مناداة البرنامج .

مثال رقم (٥,٢)

قارن حل المعادلة $y'= 2xy$ إذا كان الشرط الابتدائي $y(0)=2$ و x من 0 إلى 2 باستخدام طريقة Rung –Kutta الكلاسكية و ميرسون ، حيث إن $h=0.2$ والحل الحقيقي $y = 2e^{x^2}$.

الحل :

الأوامر التالية تُنفذ على ملف الدالة f501 :

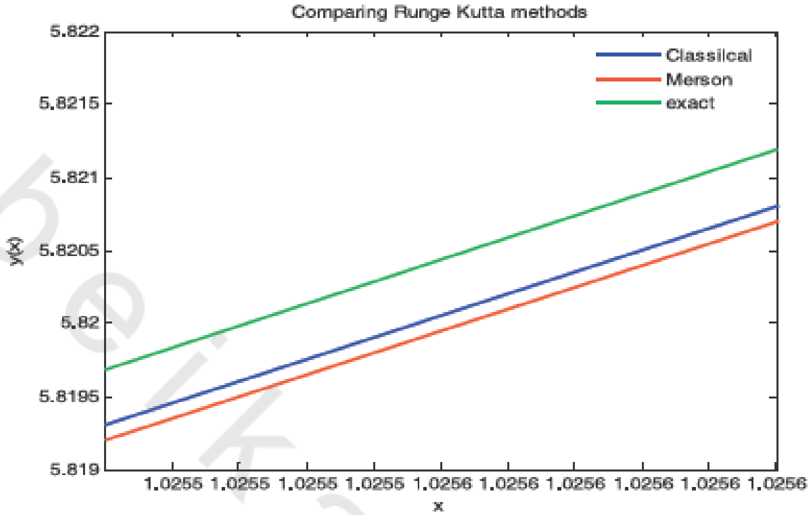
```
function yp=f501(x,y)
   yp=2*x*y;
```

ثم نكرر استخدام الخوارزمية باختيار الرقم 1 لرونج كوتا الرابعة الكلاسكية Classical RK4 أو الرقم 2 لرونج كوتا ميرسون RK Merson ونرسم النتائج الموضحة في الشكل رقم (٥,٣):

```
[tvals,yvals1]=rkgen('f501',0,2,2,0.2,1);
[tvals,yvals2]=rkgen('f501',0,2,2,0.2,2);
y=2.*exp(tvals.^2);
```

```
plot(tvals,yvals1)
hold
Current plot held
plot(tvals,yvals2,'r')
plot(tvals,y,'g')
```

الطرق تعطي نتائج متقاربة جداً، ولذلك قمنا بالتقريب باستخدام zoom في شريط مهام الشكل، عندها لاحظنا أن الحل بطريقة رونج كوتا الرابعة الكلاسكية أدق مقارنة بطرق رونج كوتا ميرسون، ولكن من عيوب كل طرق رونج كوتا كمية الحسابات الكثيرة وعدد مرات التعويض بالدالة f .



الشكل رقم (٥,٣). مقارنة طرق رونج كوتا.

(٥,٤) طريقة التخمين والتصحيح Predictor- Corrector Methods

تدعى الطرائق السابقة طرائق الخطوة الواحدة لأن التقريب يتم عن طريق نقطة واحدة سابقة، ولكن عملياً نحتاج إلى استخدام أكثر من نقطة. والطرق السابقة ضرورية للحصول على القيم الأولية، إلا أنها عموماً تتطلب جهداً للحصول على الحل العددي بالدقة المطلوبة. أما الطرائق ذات الخطوات المتعددة فهي نوعان، منها الطرائق الضمنية implicit methods والطرائق الصريحة explicit methods. نعرض هنا طرقاً تطلق عليها طرق التخمين والتصحيح Predictor-Corrector وهي توليف بين طريقة صريحة وأخرى ضمنية، حيث تخمن الطريقة الصريحة التقريب، وتصحح الطريقة الضمنية هذا التخمين. هناك طرق عديدة متعددة الخطوات تستخدم للتخمين والتصحيح نعرض إحداها وهي:

١- للتخمين: طــريقة آدمز- باشفورث Adams Bashforth ذات الخطوات الأربع و خطأ برتبة $O(h^4)$: و $i= 3,4,5,...n-1$.

$$y_0 = \alpha_0, \qquad y_1 = \alpha_1, \qquad y_2 = \alpha_2, \qquad y_3 = \alpha_3,$$
$$y_{i+1} = y_i + \frac{h}{24}(55f(t_i, y_i) - 59f(t_{i-1}, y_{i-1}) + 37f(t_{i-2}, y_{i-2}) - 9f(t_{i-3}, y_{i-3}))$$

٢- للتصحيح : طريقة آدمز- مولتون Adams-Moulton method ذات الخطوات الثلاث و خطأ برتبة $O(h^4)$ و $i= 3,4,5,...n-1$.

$$y_0 = \alpha_0, \qquad y_1 = \alpha_1, \qquad y_2 = \alpha_2,$$
$$y_{i+1} = y_i + \frac{h}{24}(9f(t_{i+1}, y_{i+1}) + 19f(t_i, y_i) - 5f(t_{i-1}, y_{i-1}) + f(t_{i-2}, y_{i-2}))$$

وفي المثال (٥,٣) التالي نستخدم Adams- Bashforth لإيجاد الحل التقريبي للمعادلة التفاضلية ، ثم نحسن الحل بطريقة Adams-Moulton method.

مثال رقم (٥,٣)

حل المعادلة $y' = -5y$ عندما $y=50$ و في الفترة $t=0$ إلى $t=6$ و h تأخذ القيم 0.2 ، 0.1 و 0.4 .

الحل :

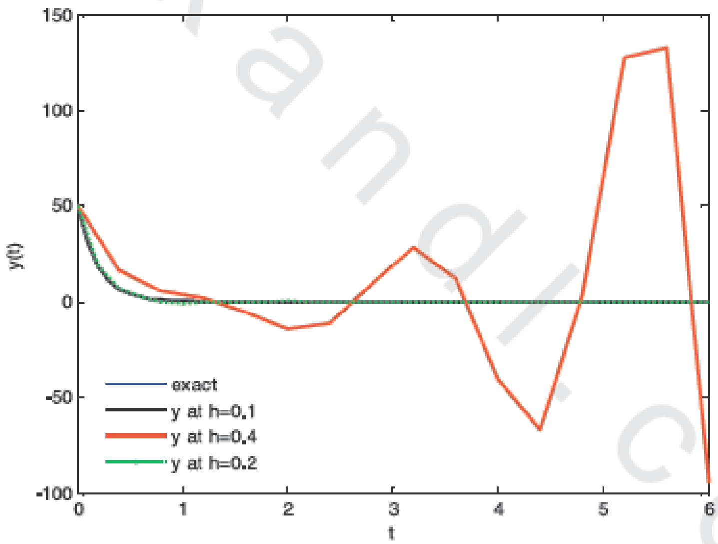
نكتب الدالة في m-file ونطبق البرنامج abm (الملحق) [15] ، مع العلم أن الحل الفعلي $y = 50e^{-5t}$.

```
function yp=f5(t,y)
yp=-5*y;
```

ثم نطبق البرنامج :

```
[t1 y1]=abm('f5',0,6,50,0.1);
[t2 y2]=abm('f5',0,6,50,0.2);
[t3 y3]=abm('f5',0,6,50,0.4);
```

من الشكل رقم (٥،٤) نلاحظ أن الحل مستقرٌ عندما h تساوي 0.1 و 0.2 ، وغير مستقر عند 0.4.



الشكل رقم (٥،٤). رسم الحلول التقريبية بخطوات مختلفة بـطريقة predictor- corrector .

(٥،٥) دوال MATLAB لحل المعادلات التفاضلية

توجد دوال جاهزة على MATLAB تقوم بحل المعادلات التفاضلية وهي : *ode45, ode15s, ode113, ode23, ode23s* و *ode23t*. نقدم شرحاً مبسطاً لبعضها في الجدول رقم

(٥,٢) مع الإشارة إلى أن اختيار الدالة المناسبة يعود لنوع المعادلة و دقة الحل المطلوب.

الجدول رقم (٢). دوال جاهزة لحل المعادلات التفاضلية.

الدالة	الوظيفة
Ode45	وهي عبارة عن نظام حل بخطوة واحدة، أي أنها تحتاج عند حساب الحل في اللحظة t_n لمعرفة القيمة التي تسبقها مباشرة t_{n-1} ويستخدم طريقة رونج كوتا ٤ او٥ الصريحة. وتأخذ الصيغة العامة الشكل [t,y] = ode45(fun,tspan,y0) ، يجب أن تكون fun على شكل الطرف الأيمن من $y'=f(x,t)$ أو $M(t,y)y'=f(x,t)$ و tspan هو مجال تغير t أما y0 فهو شرط البداية للمعادلة. وفي الأغلب تجرب كأول طريقة لحل المسائل غير الجامدة.
Ode23	طريقة أكثر فعالية وأدق من سابقتها، وهي أيضاً طريقة وحيدة الخطوة ، وتستخدم رونج كوتا ٢ أو ٣ الصريحة . الصيغة العامة : [t,y] = ode23(fun,tspan,y0).
Ode113	تستخدم طريقة الخطوات المتعددة التخمين والتصحيح ادمز- باشفولد- مولتون Adams-Bashfold-Moulton predictor corrector method.
Ode23s	تفضل هذه الدالة في حالة المسائل العنيدة و تستخدم طريقة روزنبروك المحسنة Modified Rosenbrock.
dsolve	الطريقة المتبعة لإيجاد حل معادلة تفاضلية عادية بشرط ابتدائي في البيئة الرمزية الصيغة العامة ويرمز لمعامل التفاضل بـ D dsolve('eqn1','x(t)=a',...) .

مثال رقم (٥,٤)

إذا كانت المعادلة التفاضلية :

$$2\frac{d^2x}{dt^2}+4\left(\frac{dx}{dt}\right)^2 - 2x = \cos x$$

والشرط الابتدائي $x=0$ و $dx/dt=10$ عندما $t=0$.أوجد حل المعادلة التفاضلية باستخدام ode23 وode45 في الفترة من 1 إلى 2.

الحل :

نعرف الدالة في ملف خاص m-file :

```
function fv=f54(t,x)
global p
fv=zeros(2,1);
fv(1)=x(1);
fv(2)=0.5*cos(x(2))+x(2)-(2*(x(1)^2));
```

وللوصول للحل نطبق الأوامر التالية :

```
[t1 ,x1]=ode23(@f54,[1,2],[0,10]);
[t2 , x2] = ode45(@f54,[1,2],[0,10]);
```

(٥,٥,١) دالة dsolve الرمزية

الأمر *dsolve* يقدم حلولاً لمعادلات تفاضلية عادية في بيئة syms. وتكتب الدالة بصيغة رمزية ، ويمكن أيضاً إدخال نظام من المعادلات التفاضلية.وتأخذ الصيغة العامة :

```
dsolve('eqn1','eqn2', ...)
```

يرمز في كتابة المعادلة التفاضلية على syms لمعامل التفاضل 'D' وهو بالنسبة

للمتغير المستقل 't' . إذا تلا D حرف فهو المتغير غير المستقل ، أما إذا تلاه رقم فهو تفاضل متكرر ، فالتفاضل الثاني للمتغير y(t) هو D2 y .

القيم الابتدائية تكتب على شكل معادلات مثل 'y(a)=b' أو 'Dy(a) = b' . وإذا كان عدد القيم الابتدائية أقل من عدد المتغيرات المستقلة ، فإن الحل سيحتوي على ثوابت C1, C2 . أما إذا كان الحل ليس صريحاً فإن الحل سيكون ضمنياً وستظهر جملة تنبيه لذلك. و إذا كان لم يكن هناك حل ضمني ولا حل صريح ، فتعطى جملة تنبيه وحل فارغ يتضمن كلمة *sym* . في حال كانت المعادلات التفاضلية غير خطية ، فالحل سيتضمن معادلة تفاضلية من درجة منخفضة.

مثال رقم (٥,٥)

أوجد حل المعادلة التفاضلية في البيئة الرمزية :

$$y'^2 + y^2 = 1 \qquad y(0) = 0$$

الحل :

نقوم بإدخال الأوامر التالية :

```
y = dsolve('(Dy)^2 + y^2 = 1','y(0) = 0')
y=
[sin(t ) ]
[Sin(-t)]
```

وهنا نعرض أمثلة أخرى للحالات المختلفة الأخرى :

```
>>Y = dsolve('Dy = y^2*(1-y)')
        Warning: Explicit solution could not be found; implicit solution
returned.
        Y =
        t+1/y-log(y)+log(-1+y)+C1=0
```

تنبيه بوجود حل ضمني

```
>> dsolve('Df = f + sin(t)', 'f(pi/2) = 0')
     ans =
     -1/2*cos(t)-1/2*sin(t)+1/2*exp(t)/(cosh(pi)+sinh(pi))^(1/2)
```

```
>> dsolve('D2y = -a^2*y', 'y(0) = 1, Dy(pi/a) = 0')
       ans =
      cos(a*t)
```

ويمكن حل معادلة أخرى :

```
>>syms s x p t
s=dsolve('D2x+Dx+.25*x=0','Dp+.5*p+.25*x=0','x(0)=4','Dx(0)=0','p(0)=0')
s =
  p: [1x1 sym]
  x: [1x1 sym]
>> s.x
 ans =
 exp(-1/2*t)*(4+2*t)
 >> s.p
 ans =
 -1/8*(8*t+2*t^2)*exp(-1/2*t)
```

مثال رقم (٦,٥)

أوجد حلاً للمعادلة التفاضلية :

$$\frac{d^3u}{dx^3} = u \qquad \text{with} \quad u(0)=1, \quad u'(0)=-1, \quad u''(0)=\pi$$

الحل :

```
>> dsolve('D3u=u','u(0)=1','Du(0)=-1','D2u(0)=pi','x')
 ans =
 1/3*pi*exp(x)-1/3*(1+pi)*3^(1/2)*exp(-1/2*x)*sin(1/2*3^(1/2)*x)+(1-
1/3*pi)*exp(-1/2*x)*cos(1/2*3^(1/2)*x)
```

(٥,٦) معادلات تفاضلية جزئية على MATLAB

سنتناول في هذا الجزء الحل العددي ببرامج على MATLAB لمسائل تحوي معادلة تفاضلية بمشتقات جزئية تتعلق بمسائل فيزيائية مختلفة ، وسنقتصر على المعادلات الكلاسيكية مثل المعادلات التفاضلية الجزئية الناقصية ، والتكافئية والزائدية :

$$a\frac{\partial^2 u}{\partial x^2}+b\frac{\partial^2 u}{\partial x\partial y}+c\frac{\partial^2 u}{\partial y^2}+d\frac{\partial u}{\partial x}+e\frac{\partial u}{\partial y}+fu+g=0$$

حيث إن a,b,c,d,e,f,g إما مقادير ثابتة وإما دوال في المتغيرين x و y.

تصنف المعادلة بأنها تكافئية عندما :

$$b^2-4ac=0$$

ونقول إنها ناقصية عندما :

$$b^2-4ac<0$$

وإنها معادلة زائدية عندما :

$$b^2-4ac>0$$

(٥,٦,١) المعادلات التفاضلية الجزئية الناقصية

Elliptic Partial-Differential Equations

سنقوم بدراسة المعادلة التفاضلية الجزئية الناقصية المعروفة باسم معادلة بواسون Poisson equation وهي :

$$\nabla^2 u(x,y) \equiv \frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y)$$

لكل $R \ni (x,y)$ وحيث إن:

$$(x,y) \in S, \forall u(x,y) = g(x,y)$$
$$R = \{(x,y) \mid a < x < b, c < y < d\}$$

ونرمز لحدود R بـ S مع الفرض هنا أن كلاً من f و g متصلة على مجالها، وهذا يضمن وحدانية الحل.

الطريقة المستخدمة هي تعديل لطريقة الفرق – المحدود لمسائل القيمة الحدية finite–difference method.

الخطوة الأولى هي اختيار عددين n و m وتعريف مقياس الخطوتين h و k بـ $h = \frac{(b-a)}{n}$ و $k = \frac{(d-c)}{m}$ ثم تُجزأ الفترة $[a,b]$ إلى n من الأجزاء المتساوية بطول h والفترة $[c,d]$ إلى m من الأجزاء المتساوية بطول k. باستخدام متسلسلة تايلورنصل للصيغ التالية:

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j),$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2} - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \mu_j)$$

حيث إن: $\xi_i \in (x_{i-1}, x_{i+1})$, $\mu_j \in (y_{j-1}, y_{j+1})$.

نستخدم القانونين في معادلة بواسون وذلك بالتعويض في المشتقات الجزئية بالصورة التالية:

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2}$$
$$= f(x_i, y_j) + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \mu_j)$$

لكل $i = 1,2,\ldots\ldots, n-1$ ولكل $j = 1,2,\ldots\ldots,m-1$ والتعبير عن الشروط الحدية:

$u(x_0 , y_j) = g(x_0 , y_j)$ لكل $j = 0,1\ldots\ldots m$

$u(x_n , y_j) = g(x_n , y_j)$ لكل $j = 0,1\ldots\ldots m$

$u(x_i , y_0) = g(x_i , y_0)$ لكل $i = 0,1\ldots\ldots n-1$

$u(x_i , y_m) = g(x_i , y_m)$ لكل $i = 0,1\ldots\ldots n-1$

باستخدام طريقة الفرق– المحدود واستبدال الحدود التفاضلية بالقيم التقريبية الناتجة من متسلسلة تايلور نحصل على معادلة الفــرق حيث إن $x_i=a+ih, y_j=c+jk$ ذات خطأ قطع برتبة $O(k^2+h^2)$:

$$2[(h/k)^2+1]u_{i,j}-(u_{i+1,j}+u_{i-1,j})-(\frac{h}{k})^2(u_{i,j-1}+u_{i,j+1})=-h^2 f(x_i,y_j)$$

وبذلك تُحول المعادلة التفاضلية إلى نظام من المعادلات الخطية ، والذي يعد حله هو الحل العددي للمعادلة التفاضلية الجزئية الأصلية. نستخدم برنامج MATLAB لحل هذا النظام حسب الطرق الواردة في الفصل الثالث. نطبق خوارزمية معادلة بواسون للفرق المحدود (الملحق) في المثال رقم (٥،٥).

مثال رقم (٥،٧)

استخدم MATLAB لتقريب حل المعادلة التفاضلية الناقصية ، وقارن النتائج مع الحل الفعلي $u(x,y)=(x-y)^2$. :

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=4, \quad 0<x<1, \quad 0<y<2;$$
$$u(x,0)=x^2, \quad u(x,2)=(x-2)^2, \quad 0\le x\le 1;$$
$$u(0,y)=y^2, \quad u(1,y)=(y-1)^2, \quad 0\le y\le 2;$$

الحل :

نطبق البرنامج *poison* (الملحق) التي تطلب إدخال البيانات الضرورية حسب التالي :

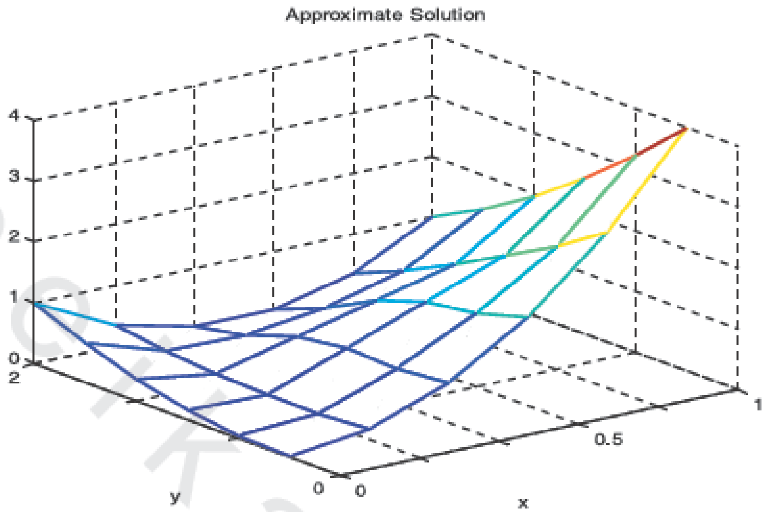
```
>> poison

The Elliptic equation of the form
d^2u/dt^2 + d^2u/dx^2 = f(x,y)  a<x<b,  c<y<d
Subject to the boundary conditions

u(a,y)=g(a,y)        c<y<d
```

```
u(b,y)=g(b,y)
u(x,c)=g(x,c)        a<x<b
u(x,d)=g(x,d)
```

```
the number of grid sections for the x variable; n= 5
enter the number of grid sections for the y variable; m = 5
enter the maximum number of iterations 50
enter the right end point of the range of x ; a= 0
enter the left end point of the range of x ; b= 1
enter the right end point of the range of y ; c= 0
enter the left end point the range of y ; d= 2
enter the tolerance ; tol=.0005
enter the function f(x,y)=4
enter the exact solution e(x,y)=(x-y).^2
enter boundary condition u(x,c)=(x).^2
enter boundary condition u(x,d)=(x-2).^2
enter boundary condition u(a,y)=(y).^2
enter boundary condition g(b,y)=(1-y).^2
```

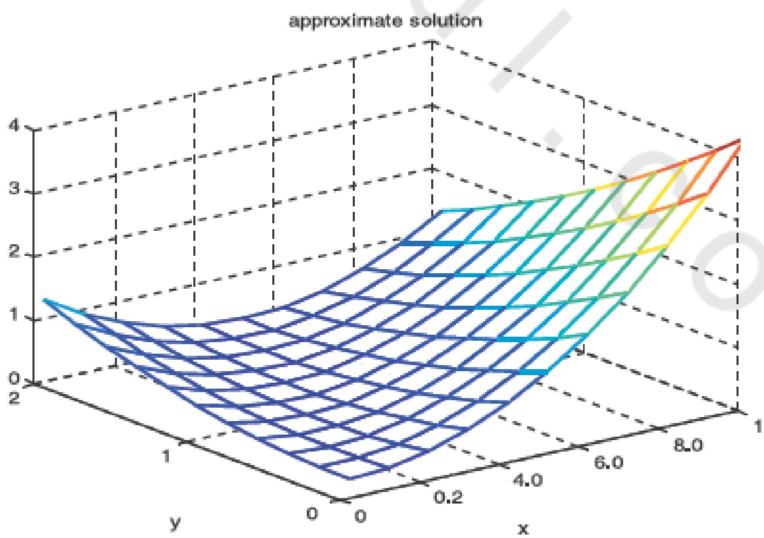
يظهر لنا الحل الفعلي و التقريبي (الشكل رقم ٥,٥) ومربع الخطأ التام total squared error :

```
The exact solution is e =
 0.1600   0.6400   1.4400   2.5600   4.0000        0
   0.0400   0.0400   0.3600   1.0000   1.9600   3.2400
 0.1600   0.6400   1.4400   2.5600      0.1600        0
   0.3600   0.0400   0.0400   0.3600   1.0000   1.9600
0.1600   0.6400   1.4400      0.6400   0.1600        0
   1.0000   0.3600   0.0400   0.0400   0.3600   1.0000

The approximated solution is w =
0.1600   0.6400   1.4400   2.5600   4.0000        0
   0.0400   0.1081   0.5644   1.2043   2.0279   3.2400
   0.1600   0.1192   0.6032   1.0830   1.5589   2.5600
   0.3600   0.1592   0.4833   0.8032   1.1190   1.9600
   0.6400   0.2282   0.2046   0.3645   0.7080   1.4400
   1.0000   0.3600   0.0400   0.0400   0.3600   1.0000

error =   1.0280
```



الشكل رقم (٥,٥). الحل الناتج لمعادلة بواسون m=n=5 .

كلما كبرت قيمة m و n فإن الحل يتحسن ، والرسم يصبح أدق وهذا ما يتضح من الرسم (الشكل رقم ٥,٦) عند m=n=10.



الشكل رقم (٥,٦). الحل الناتج لمعادلة بواسون بقيم m=n=10 .

(٥,٦,٢) المعادلات التفاضلية الجزئية التكافئية

Parabolic Partial-Differential Equations

نعرض معادلة الحرارة أو الانتشار، وهي معادلة تفاضلية جزئية تكافئية، تُستخدم في نمذجة انتشار درجة الحرارة في قضيب معدني طوله l سم ، وتكتب المسألة والشروط الابتدائية الحدية كالتالي :

$$\frac{\partial u}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t), \qquad 0 < x < l, \qquad t > 0$$

تحت الشروط : $u(0,t) = 0, \qquad u(l,t) = 0, \qquad t > 0$

$$u(x,0) = f(x), \qquad 0 \le x \le l$$

حيث إن α^2 يمثل معامل الانتشار ولإيجاد درجة الحرارة عند أي نقطة x وزمن t في قضيب يكون في درجة حرارة صفر عند النهايات $x=l$ و $x=0$ و بدرجة حرارة ابتدائية في القضيب $f(x)$. يتضمن الأسلوب الذي نستخدمه لتقريب حل هذه المسألة فروقاً محدودة ، وهي مشابهة للطريقة المستخدمة لحل المعادلات التفاضلية الجزئية الناقصية.

نختار أولاً ثابتي الشبكة h و k بشرط أن يكون $m = \frac{l}{h}$ عدداً صحيحاً. وبذلك نحصل على طريقة الفرق باستخدام متسلسلة تايلور، ويمكن استخدام برنامج MATLAB لحل معادلة الحرارة بخوارزمية الفرق التراجعي Backward-difference method مع مراعاة شروط الاستقرار وكون خطأ القطع برتبة $O(k+h^2)$:

$$(1+2\alpha^2 k/h^2)u_{i,j}-(\alpha^2 k/h^2)(u_{i+1,j}+u_{i-1,j})=u_{i,j-1}$$

أو يمكننا استخدام طريقة كرانك– نيكلسون Crank-Nicholson method المستقرة بدون شروط وذات خطأ قطع من الرتبة $O(k^2+h^2)$:

$$\frac{u_{i,j+1}-u_{i,j}}{k}-\frac{\alpha^2}{2}\left[\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+\frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{h^2}\right]=0$$

مثال رقم (٥,٨)

قرب حل المعادلة التفاضلية الجزئية الآتية مستخدماً خوارزمية الفرق التراجعي:

$$\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}=0,\qquad 0<x<2,\qquad 0<t;$$

بالشروط: $u(0,t)=u(2,t)=0, \qquad t>0$

$u(x,0)=\sin\frac{\pi}{2}x, \qquad 0\le x\le 2.$

باستخدام n=100 ، t=0.1 و m=10 وقارن النتائج بالحل الحقيقي:

$$u(x,t)=e^{-(\pi^2/4)t}\sin(\frac{\pi}{2})x.$$

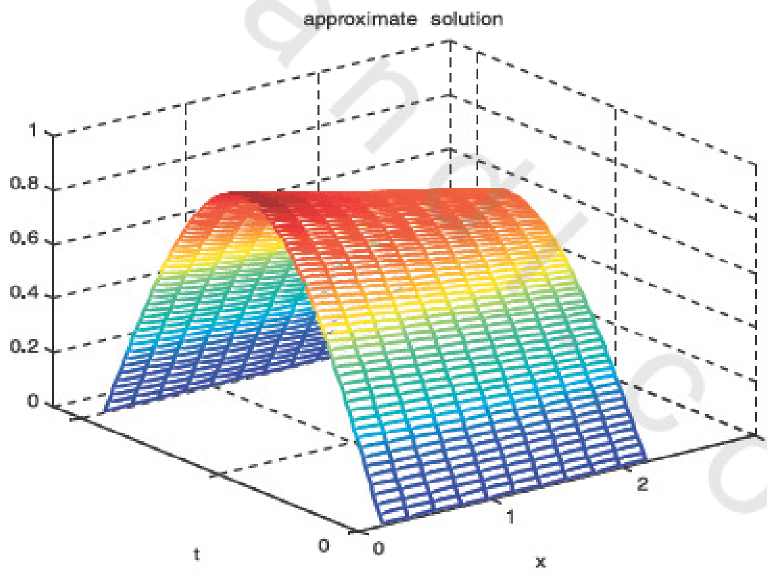
الحل:

الحل باستخدام برنامج *heat* (الملحق) ينتج الشكل رقم (٥,٧):

```
>> heat
The Parabolic equation of the form
d^2u/dt^2 - (alpha^2)* d^2u/dx^2 =0  0<x<l,  0<t<T
```

Subject to the boundary conditions
u(0,t)=T1
u(1,t)=T2   0<t<T
u(x,0)=f(x)   0<=x<=l

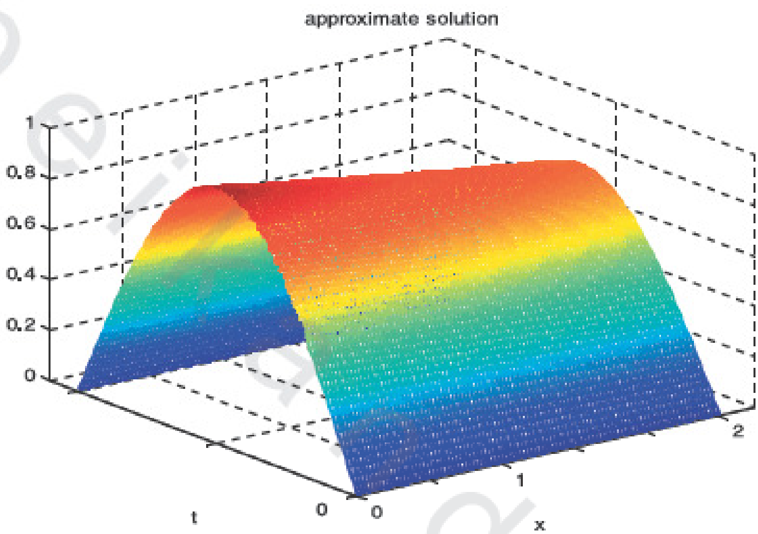
enter the number of grid sections for the x variable; m= 10
enter the number of grid sections for the t variable; n = 100
enter the end point of the range for x ; l= 2
enter the end point of the range for t ; T= .1
enter the exact solution e(x,t)=exp(-t.(pi^2/4)).*sin((pi/2).*x)*
enter the constant alpha = 1
enter the constant T1 = 0
enter the constant T2 = 0
*enter boundary condition u(x,0)=f(x)=sin((pi/2).*x)*



الشكل رقم (٥,٧). الحل الناتج لمعادلة الحرارة بقيم n=10 , m= 100 .

ولكن مع مراعاة شرط الاستقرار stability condition $\alpha^2(k/h^2) < 0.5$ أما عند استخدام كرانك– نيكلسون فإن الحل يكون أكثر دقة والطريقة مستقرة بدون شروط

كما في الشكل رقم (٥,٨)، باستخدام 100 =m و n=100 التي قد لا تكون مستقرة بخوارزمية الفرق التراجعي.



الشكل رقم (٥,٨). حل تقريبي لمعادلة الحرارة بقيم m= 100, n=100 بكرانك نيكلسون.

(٥,٦,٣) المعادلات التفاضلية الجزئية الزائدية

Hyperbolic Partial-Differential Equations

نتطرق للمعادلة التفاضلية الموجية كمثال لمعادلة تفاضلية جزئية زائدية وتعطى بالمعادلة التفاضلية الجزئية التالية :

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t) = 0, 0 < x < l, \qquad t>0$$

تحت الشروط :

$$u(0,t)=u(l,t)=0\,, \qquad t>0$$
$$u(x,0)=f(x), \qquad 0\le x\le l$$
$$\frac{\partial u}{\partial t}(x,0)=g(x), \qquad 0\le x\le l$$

ويستخدم برنامج MATLAB لحل المعادلة الموجية خوارزمية الفرق المحدود للمعادلة الموجية وهي بخطأ قطع برتبة $O(h^2+k^2)$ وشرط استقرار $\alpha\, k/h \le 1$ ، ويتم الحصول على طريقة الفرق باستخدام الفرق المركزي للمشتقات الجزئية الثانية لنصل الى معادلة الفرق الصريحة :

$$\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{k^2}-\alpha^2\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}=0$$

مثال رقم (٥,٩)

أوجد حل المعادلة :

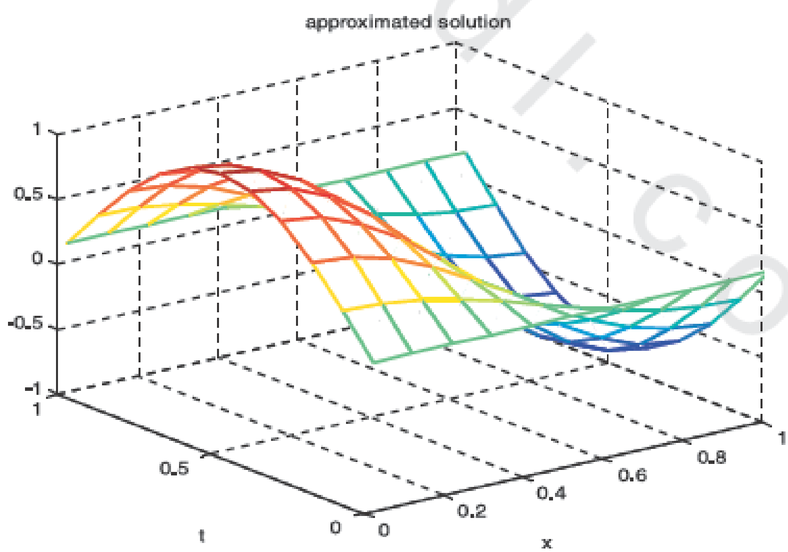
$$\frac{\partial^2 u}{\partial t^2}-\frac{\partial^2 u}{\partial x^2}=0, \quad 0<x<1, \qquad t>0$$
$$u(0,t)=u(1,t)=0, \qquad t>0$$
$$u(x,0)=\sin\pi x\,, \qquad 0\le x\le 1,$$
$$\frac{\partial u}{\partial t}(x,0)=0, \qquad 0\le x\le 1,$$

مستخدماً $n=10$,$m=10$ و $T=1$ وقارن بالحل الحقيقي للمعادلة $u(x,t)=\cos(\pi t)\sin(\pi x)$.

الحل :

نطبق البرنامج *wave* (الملحق) لينتج الحل التقريبي (الشكل رقم ٥,٩):

```
>> wave
The Hyperbolic equation of the form

d^2u/dt^2 - (alpha^2)*d^2u/dx^2 = 0  0<x<l,  0<t<T

Subject to the boundary conditions

u(0,t)=u(l,t)=0        0<t<T

u(x,0)=f(x)

du/dt=g(x)        0<x<l

enter the number of grid sections for the t variable; n=  10
enter the number of grid sections for the x variable; m =  10
enter the end point of the range for x; l=  1
enter the end point of the range for t; T=  1
enter the constant alpha=  1
enter the boundary condition f(x)=sin(pi*x)
enter the boundary condition g(x)=0
enter the exact solution e(x,t)=cos(pi*t).*sin(pi*x)
```



الشكل رقم (٥،٩). حل تقريبي لمعادلة الموجه بقيم m= 10,n=10 .

الطرائق التي قُدمت في تطبيقات المعادلات التفاضلية الجزئية كانت مستقرة أو مشروطة الاستقرار وكلها طرائق صريحة ولكن توجد طرائق أخرى ضمنية ذات استقرار بدون شروط ويمكن رؤية دراسة لهذه الطرائق [22] .

(٥,٧) تمارين

حل المعادلات التفاضلية التالية باستخدام دالة جاهزة في MATLAB أو ببرنامج m-file :

١-

$$y'' = -(y')^2 - y + \ln x, \qquad 1 \le x \le 2$$
$$y(1) = 0,$$
$$y(2) = \ln 2$$

٢-

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 \le x \le \pi \qquad t > 0$$
$$u(0,t) = u(1,t) = 0, \qquad t > 0$$
$$u(x,0) = \sin x \qquad 0 \le x \le \pi$$

٣-

$$\frac{\partial u}{\partial t}(x,0) = 0, \qquad 0 \le x \le \pi$$
$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 < x < 2, \qquad t > 0$$
$$0 \le x \le 2. \quad 0 < t; \; u(0,t) = u(2,t) = 0,$$

٤- أوجد الحل التقريبي بطريقة عددية مناسبة وقارن بالحل الفعلي.

أ) $$2\frac{d^2x}{dt^2}+4\left(\frac{dx}{dt}\right)^2-2x=\cos x$$

حيث إن $dy/dx=10$ و $x=0$ عندما $t=0$

ب) $-y''+y=x$ بحيث إن $0<x<1$ $y(0)=1$ و $y(1)=1+e^{-1}$

جـ) $y'=\sin t+e^{-t}$ $\quad y(0)=0,$ $\quad 0\le t\le 1$

د) $y'=t^2$ $\quad y(0)=0,$ $\quad 0\le t\le 2$

٥- استخدم الخوارزمية المناسبة لتقريب الحل:

أ) $y'=1-y$ $\quad y(0)=0,$ $\quad 0\le t\le 1$

ب) $y'=-y+t+1$ $\quad y(0)=2,$ $\quad 0\le t\le 5$

جـ) $y''=-(y')^2-y+\ln x$ $\quad y(1)=0,\ y(2)=\ln 2$ $\quad 1\le x\le 2$

د) $y''=4(y-x)$ $\quad y(0)=0,\ y(1)=2$ $\quad 0\le x\le 1$